QtCon BR

Qt For Robots: Interfaces, Sensores e Simuladores

Patrick Pereira
patrick@bluerobotics.com
patrickjp@kde.org

whoami?

Patrick Pereira / Electronic Eng.

 @patrickelectric

 @patrickelectric

 @patrickelectric

 https://patrickelectric.work

Work

Software Engineer

Software development for ROVs, sensors and user applications.

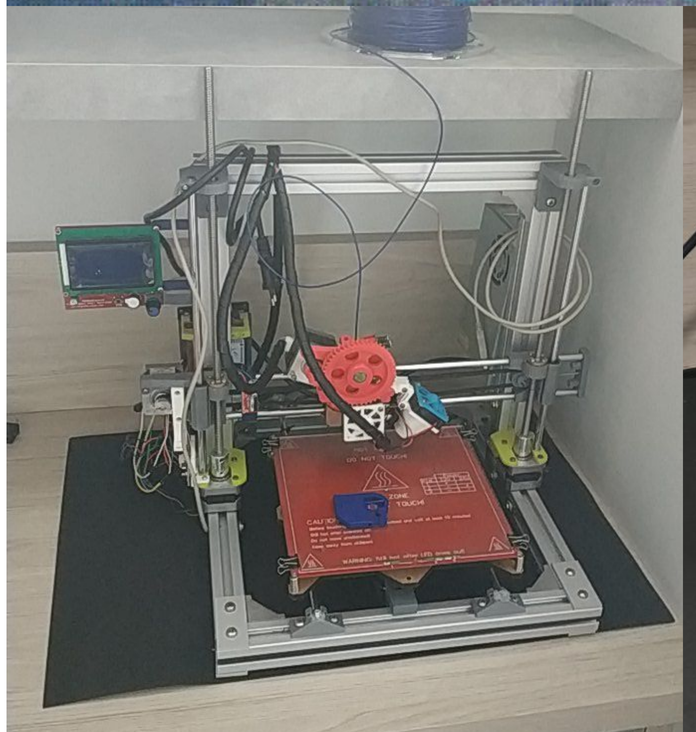
BlueRobotics

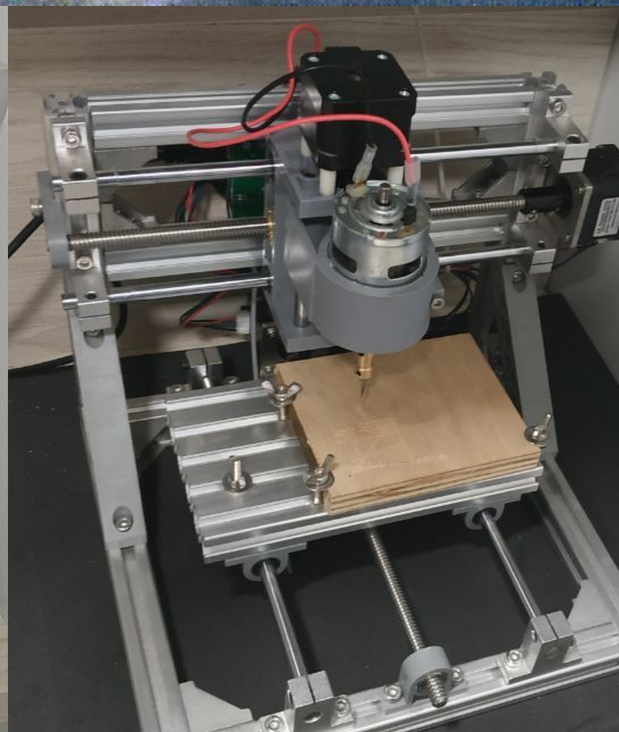
Hobby

Programmer

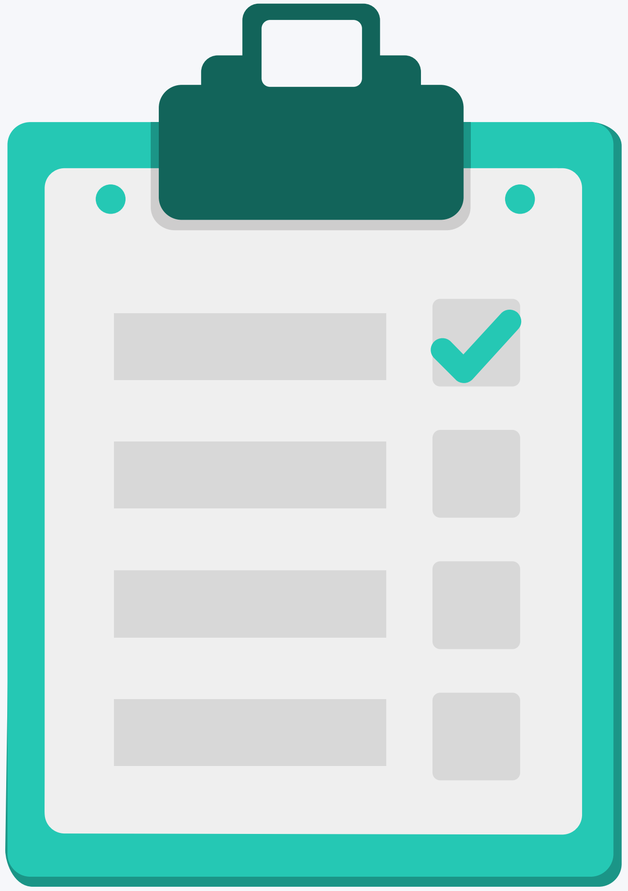
Contributions with AtCore, Atelier and random KDE projects.

atcore



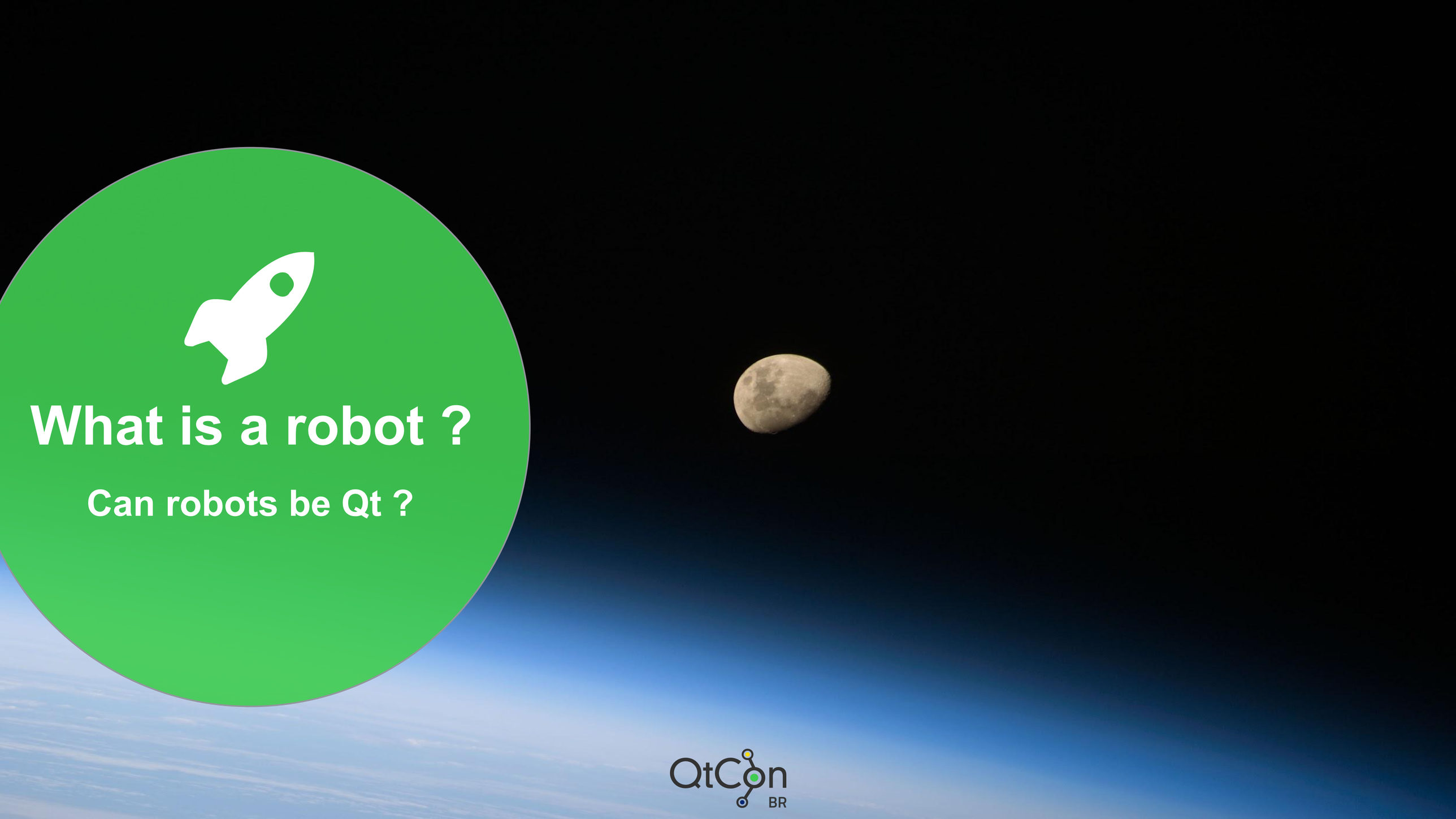


Our Agenda



0. Introduction.
1. Data visualization.
2. Sensor integration.
3. Control.
4. 2D simulations.
5. 3D simulations.

QtCon BR



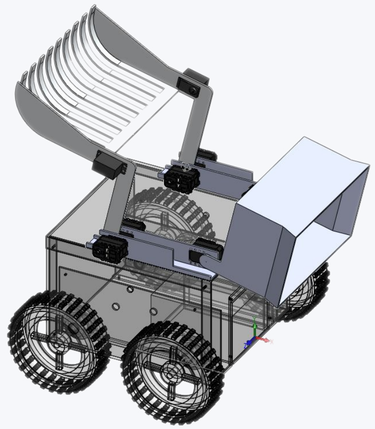
What is a robot ?

Can robots be Qt ?

QtCon BR

Examples

ROBOTA

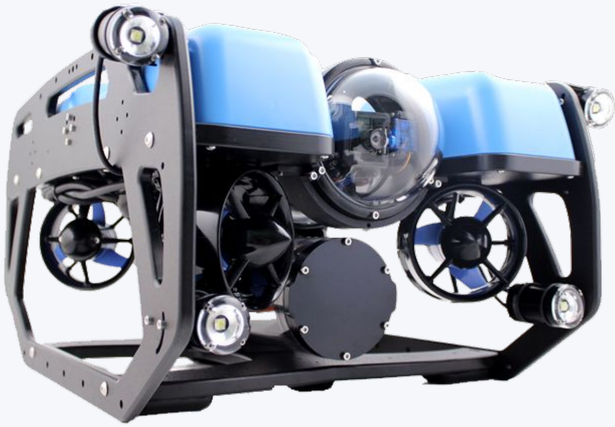


PROVANT
UFSC UFMG

HORUS
AERONAVES
DRONE MAPPING SOLUTIONS

Source: horusaeronaves.com

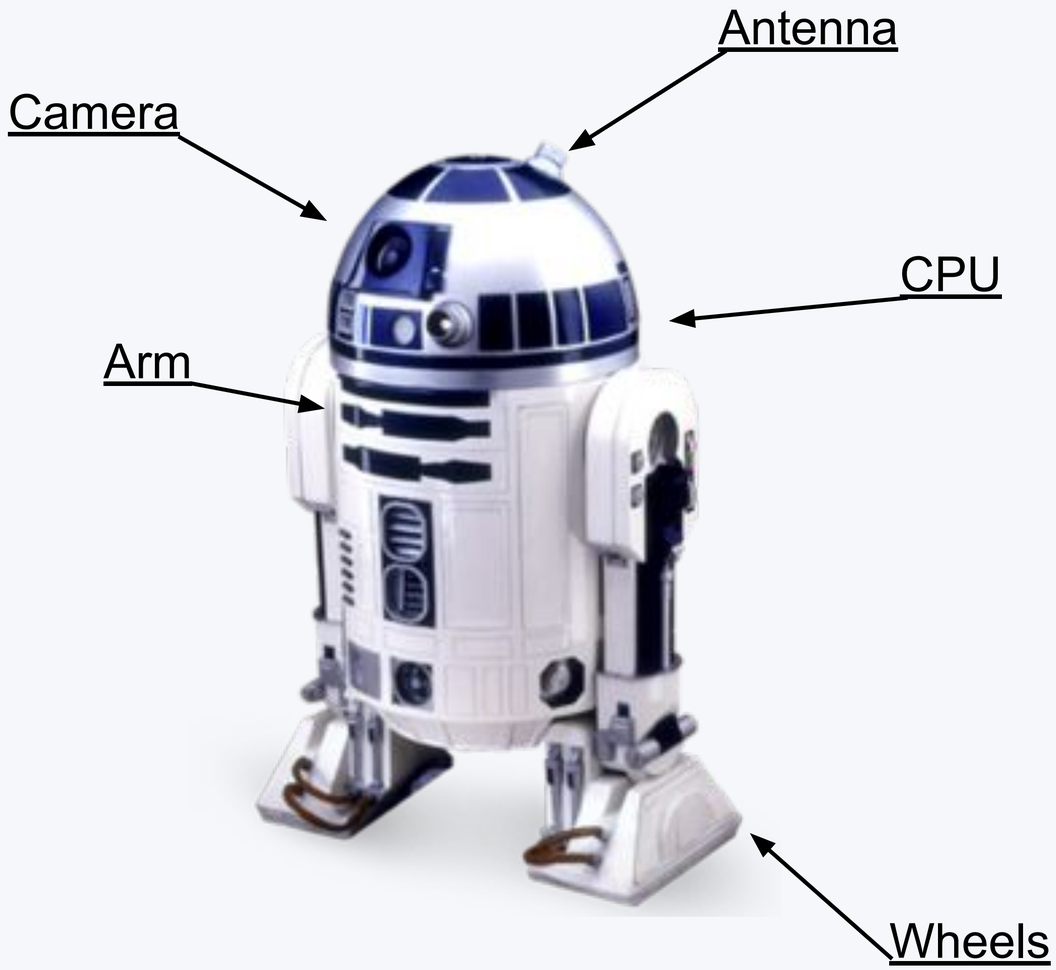
BlueRobotics



Source: bluerobotics.com

QtCon BR

Show me what you got



QtCon BR

source: Wikipedia

Sensors

USB

I2C

SPI

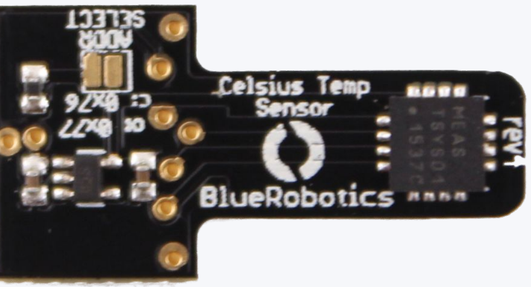
UDP

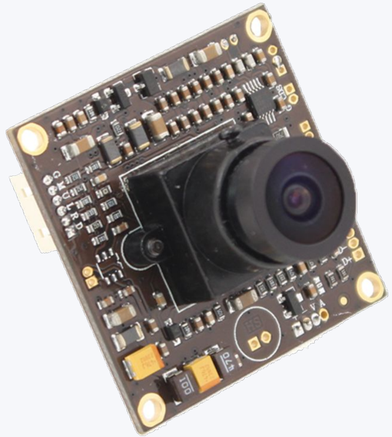
RS485

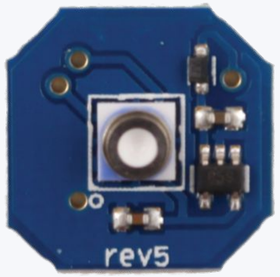
Serial

CAN

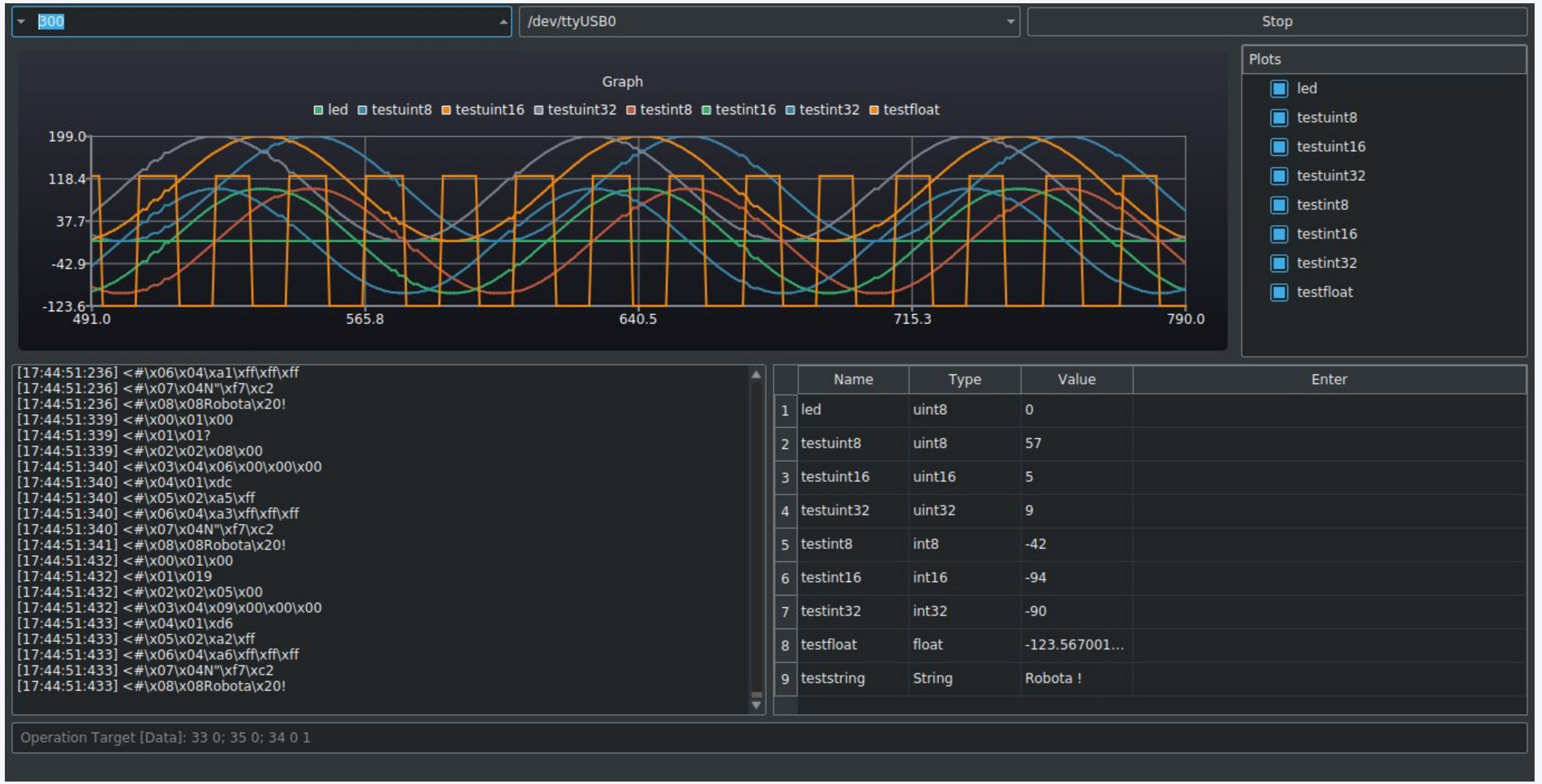
TCP





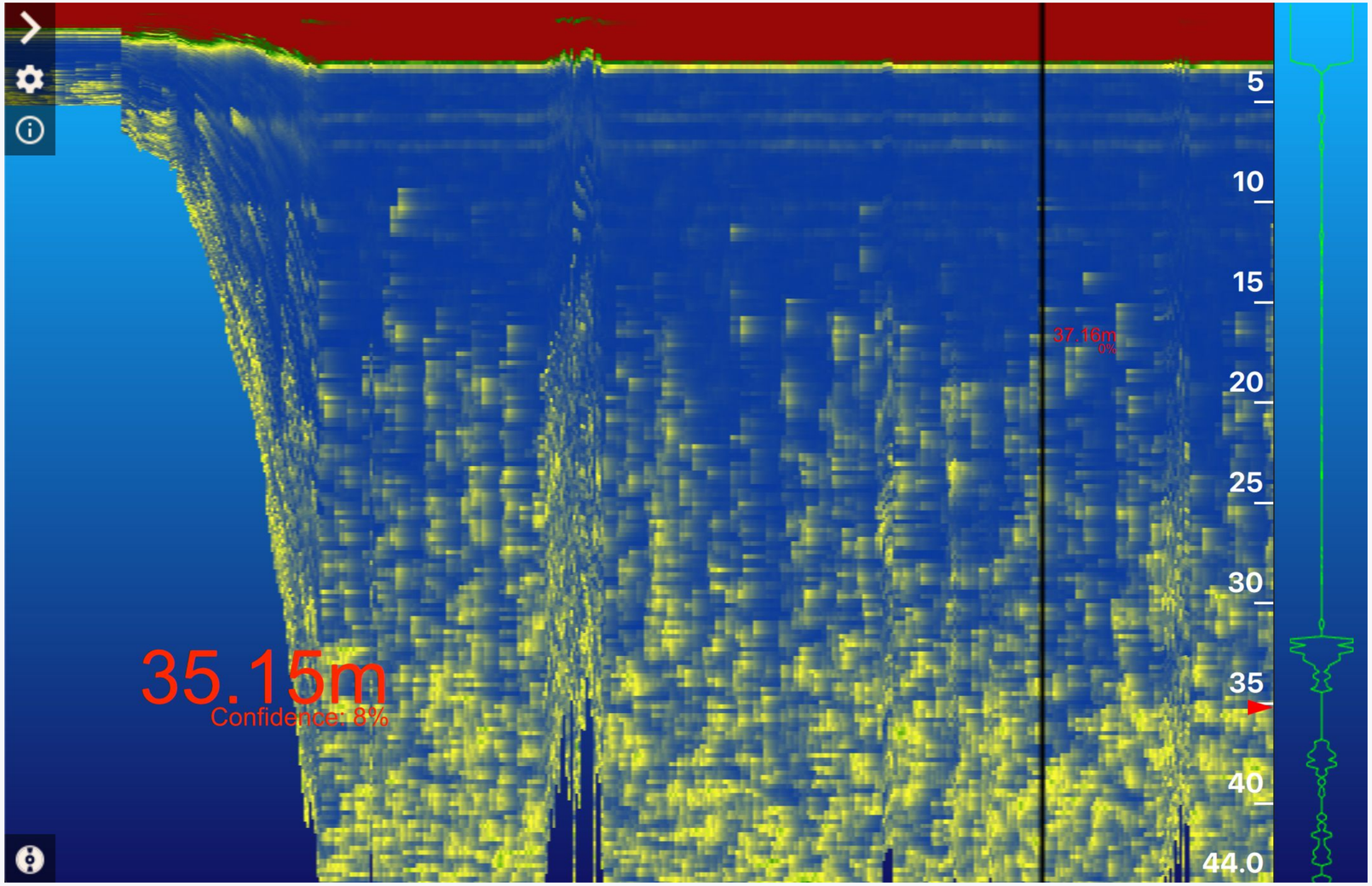


Exposer-GUI



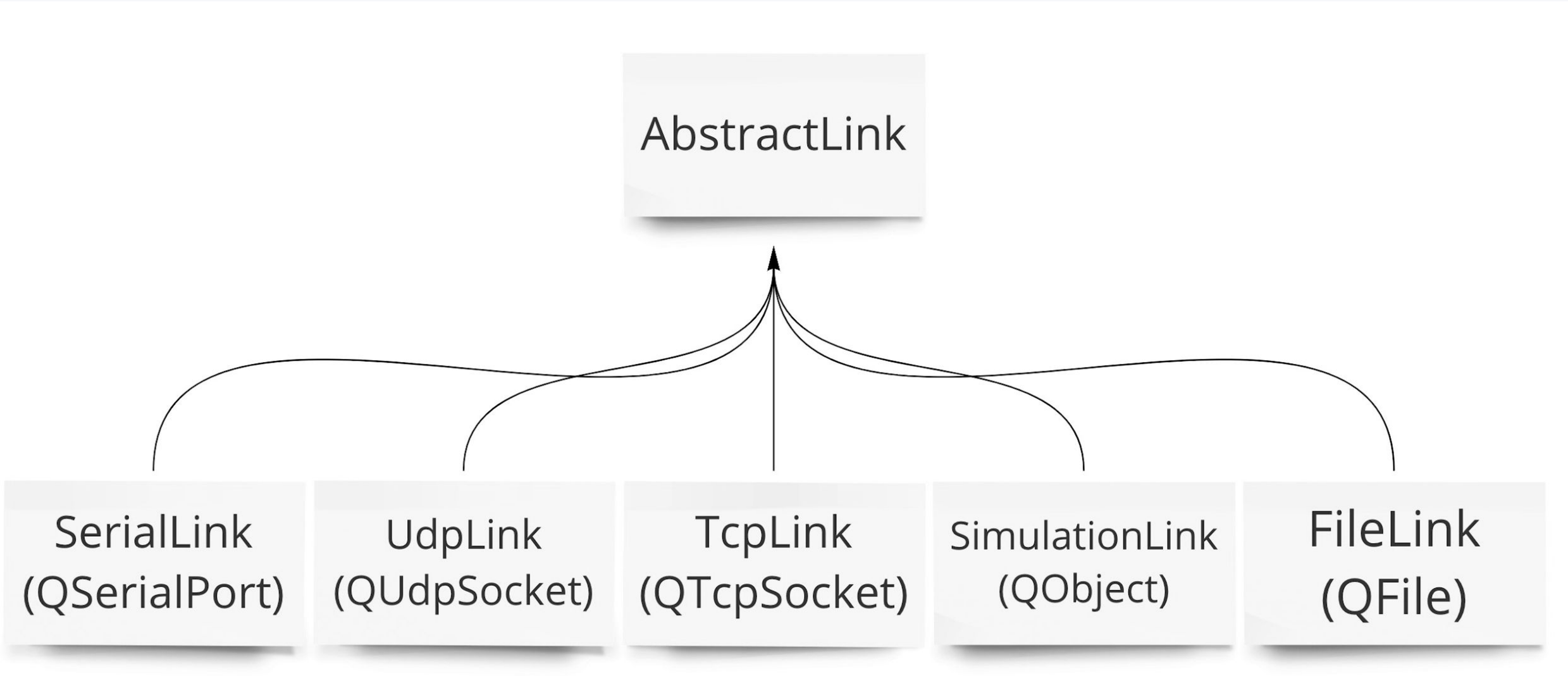
QtCon BR

PingViewer [Demo]



QtCon BR

AbstractLink



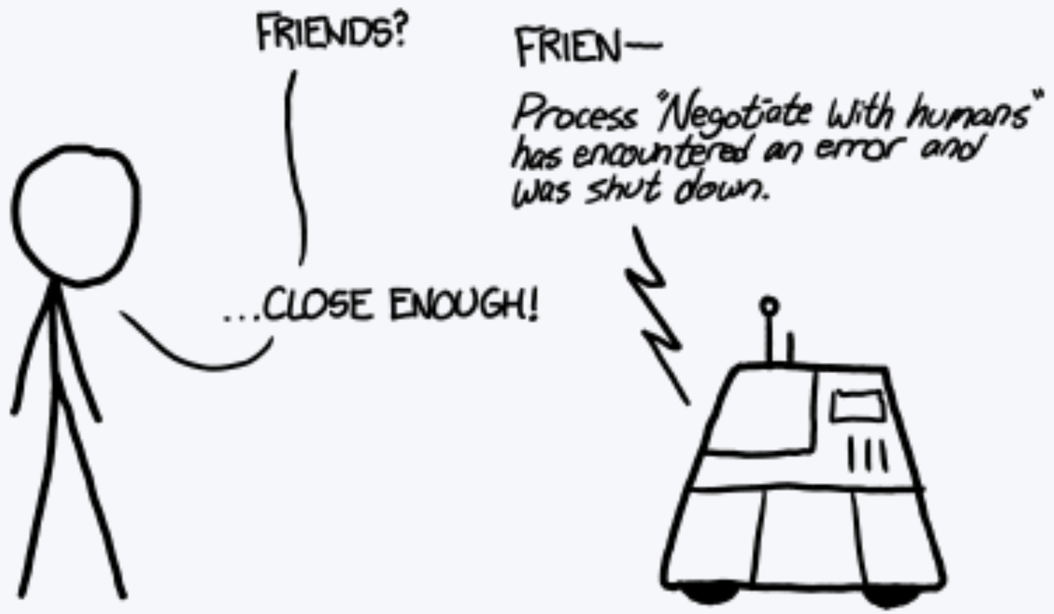
AbstractLink

```
1.  //link.cpp
2.  Link::Link(LinkType linkType, QString name, QObject* parent)
3.      : QObject(parent)
4.      , _abstractLink(nullptr)
5.  {
6.      switch(linkType) {
7.      case LinkType::File :
8.          _abstractLink = new FileLink();
9.          break;
10.     case LinkType::Serial :
11.         _abstractLink = new SerialLink();
12.         break;
13.     case LinkType::Udp :
14.         _abstractLink = new UDPLink();
15.         break;
16.     case LinkType::PingSimulation :
17.         _abstractLink = new PingSimulationLink();
18.         break;
19.     default :
20.         qCCritical(PING_PROTOCOL_LINK) << "Link not available!";
21.         return;
22.     }
23. ...
```

```
1.  //sensor.cpp
2.  ...
3.  QSharedPointer<Link> _link;
4.  …
5.  AbstractLink* link() { return _link.data() ? _link->self() : nullptr; };
6.  Q_PROPERTY(AbstractLink* link READ link NOTIFY linkUpdate)
7.  …
8.  if(!link()->isOpen())
9.  …
10. if(!link()->isWritable())
11. …
12. if(link()->type() == LinkType::File)
13. …
14. connect(link(), &AbstractLink::newData, _parser, &Parser::parseBuffer);
15. …
16. SerialLink* serialLink = dynamic_cast<SerialLink*>(link());
17. ...
```

QtCon BR

Control



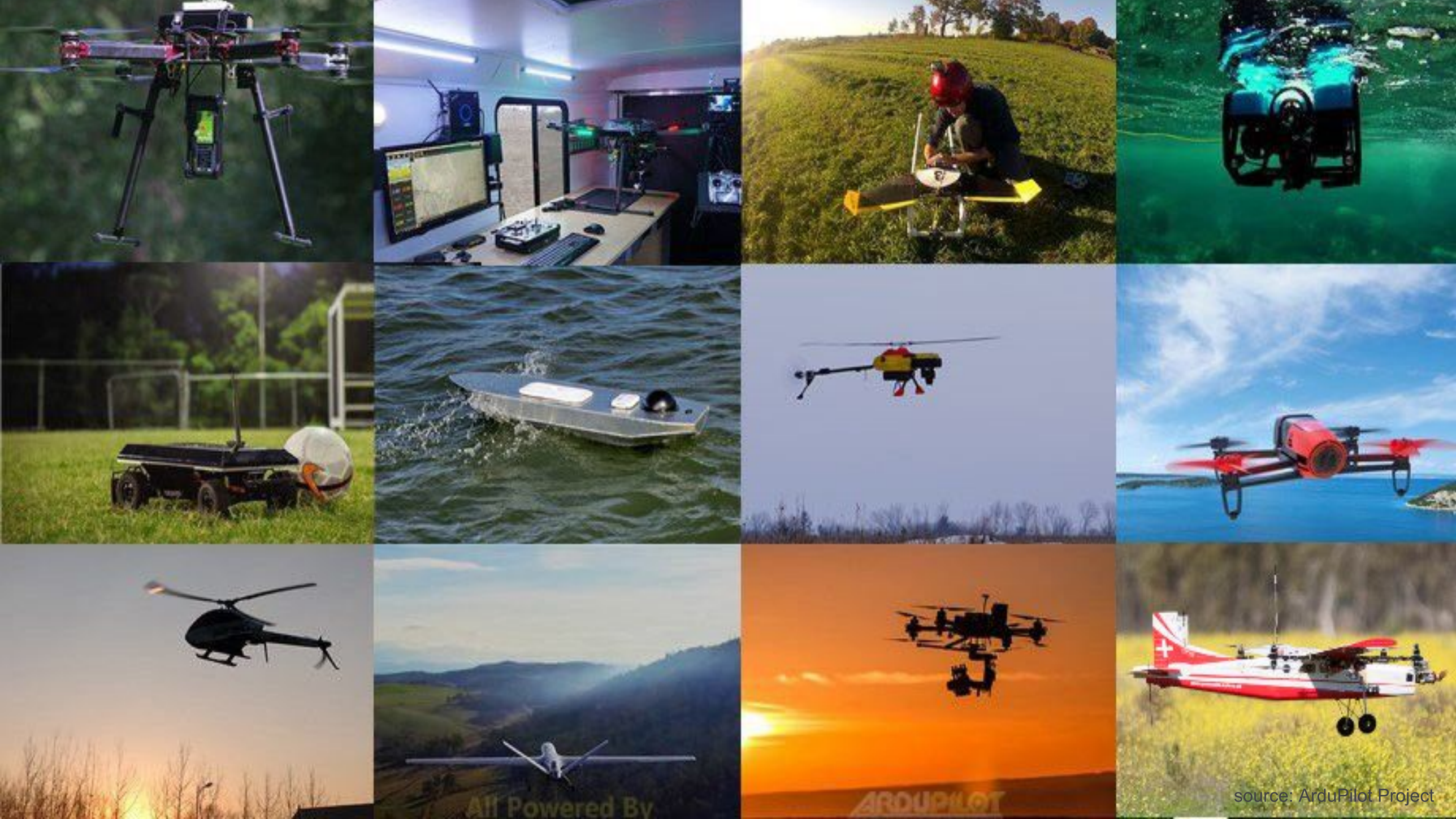
QtCon BR

source: XKCD

Planes, Rovers, Boats and Submarines

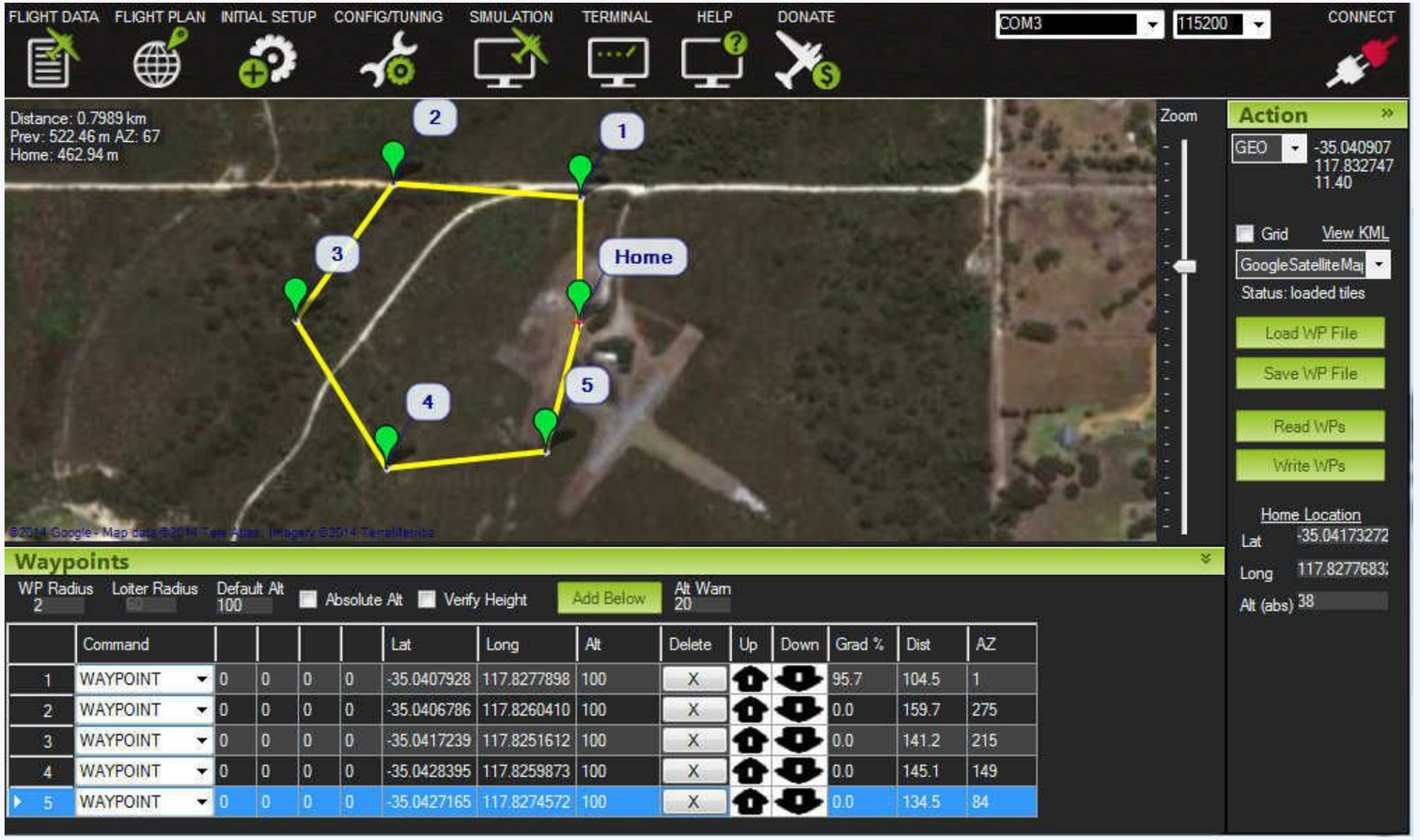
The most advanced, full-featured and reliable open source autopilot software available. It is the only autopilot software capable of controlling any vehicle system imaginable, from conventional airplanes, multirotors, and helicopters, to boats and even submarines.

QtCon BR


source: ArduPilot Project

Mission Planner

- Windows only
- C#
- Slow user input
- Older UI

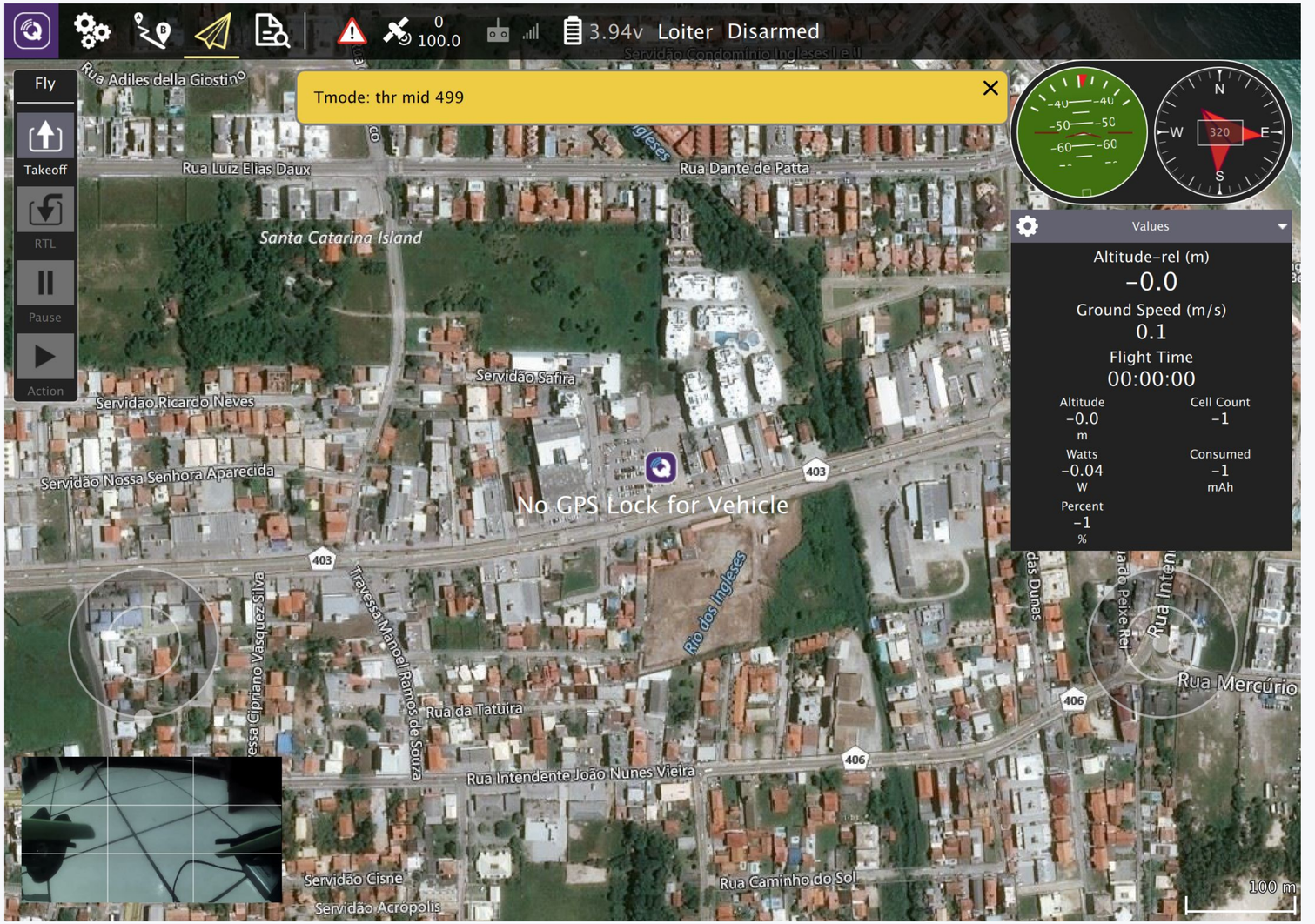


QtCon BR

source: ardupilot.org

QGroundcontrol [Demo]

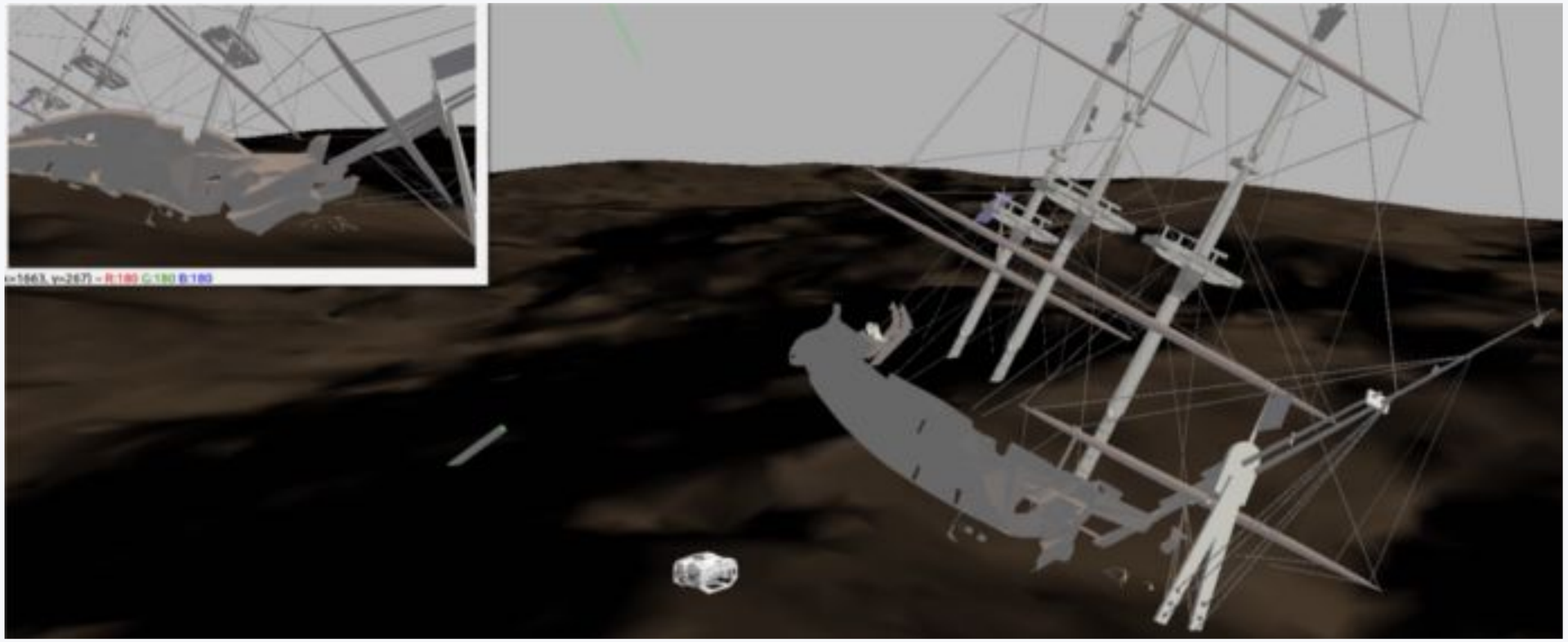
- Windows, Linux, Mac, Android and iOS
- C++
- Faster user input
- Modernish UI



QtCon BR

Simulation

- Gazebo
- V-REP
- ARGoS
- Microsoft Robotics
- ...

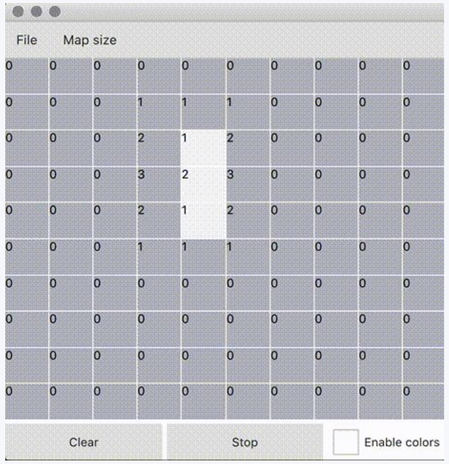


Gazebo with BlueROV simulation

Simulation

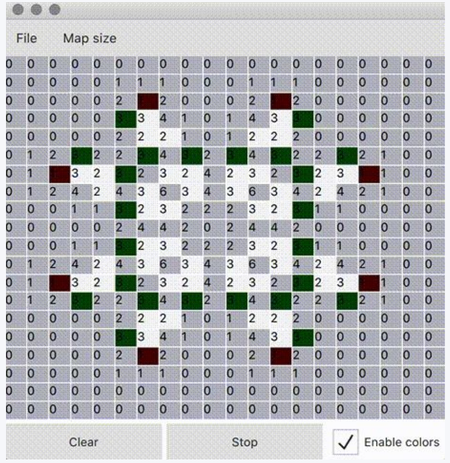
Behavior

- ✓ Development
- ✓ Debug
- ✓ Validation



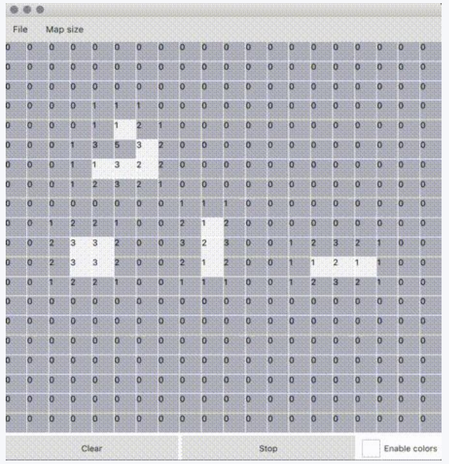
Desires

- ✓ Changes based on last state.



Interactions

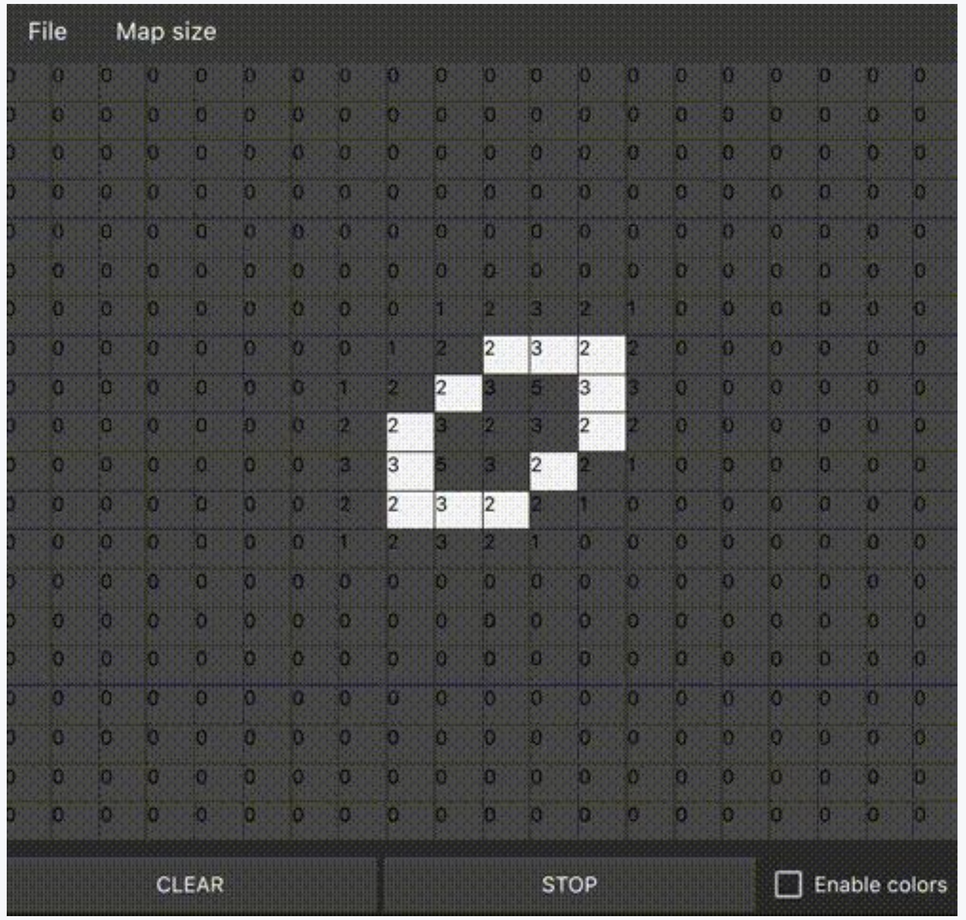
- ✓ Interaction between models
- ✓ Interaction of model with ambient
- ✓ Simulation result and consequences



QtCon BR

2D Simulation

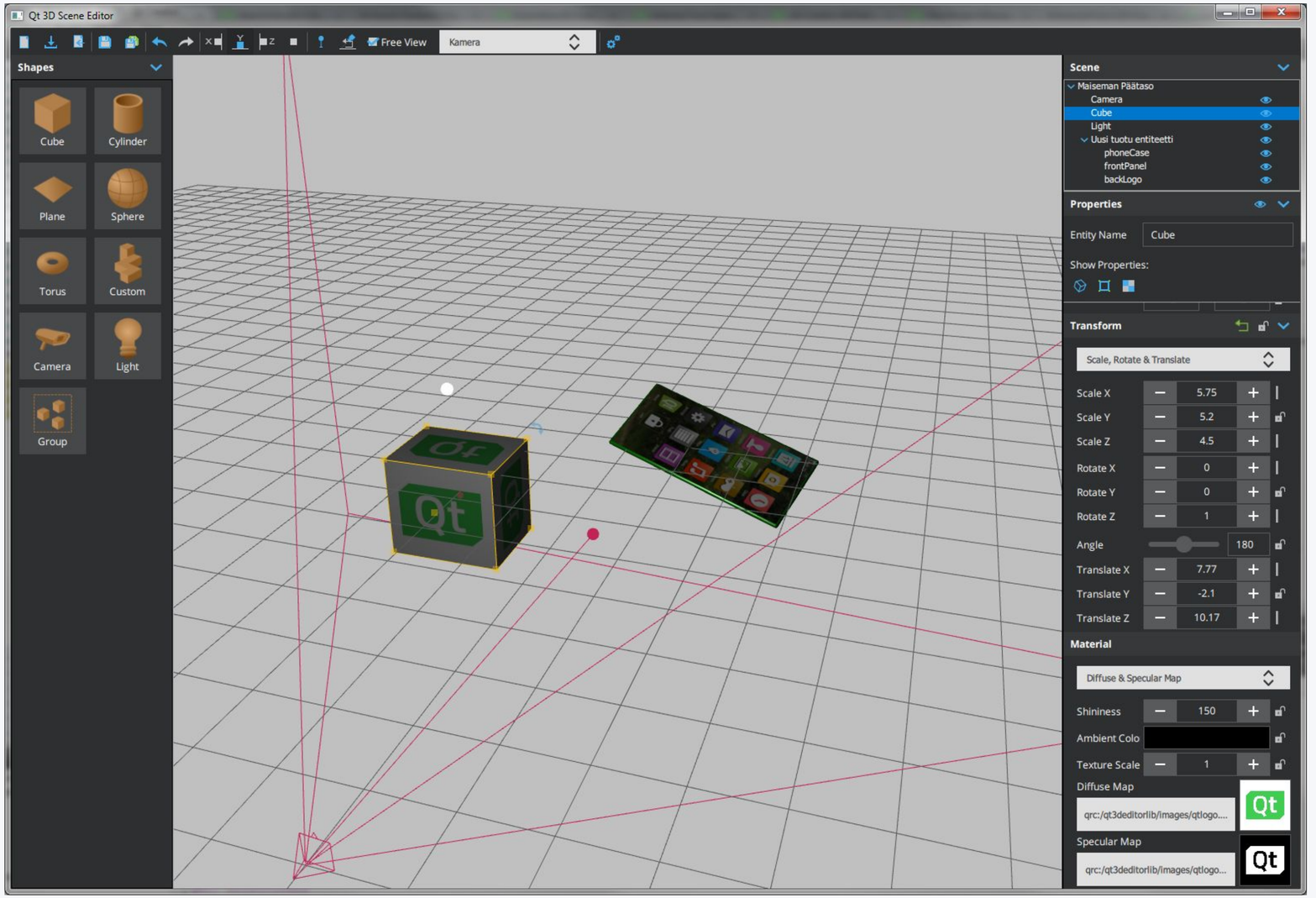
SLAM



Conway's GoL

QtCon BR

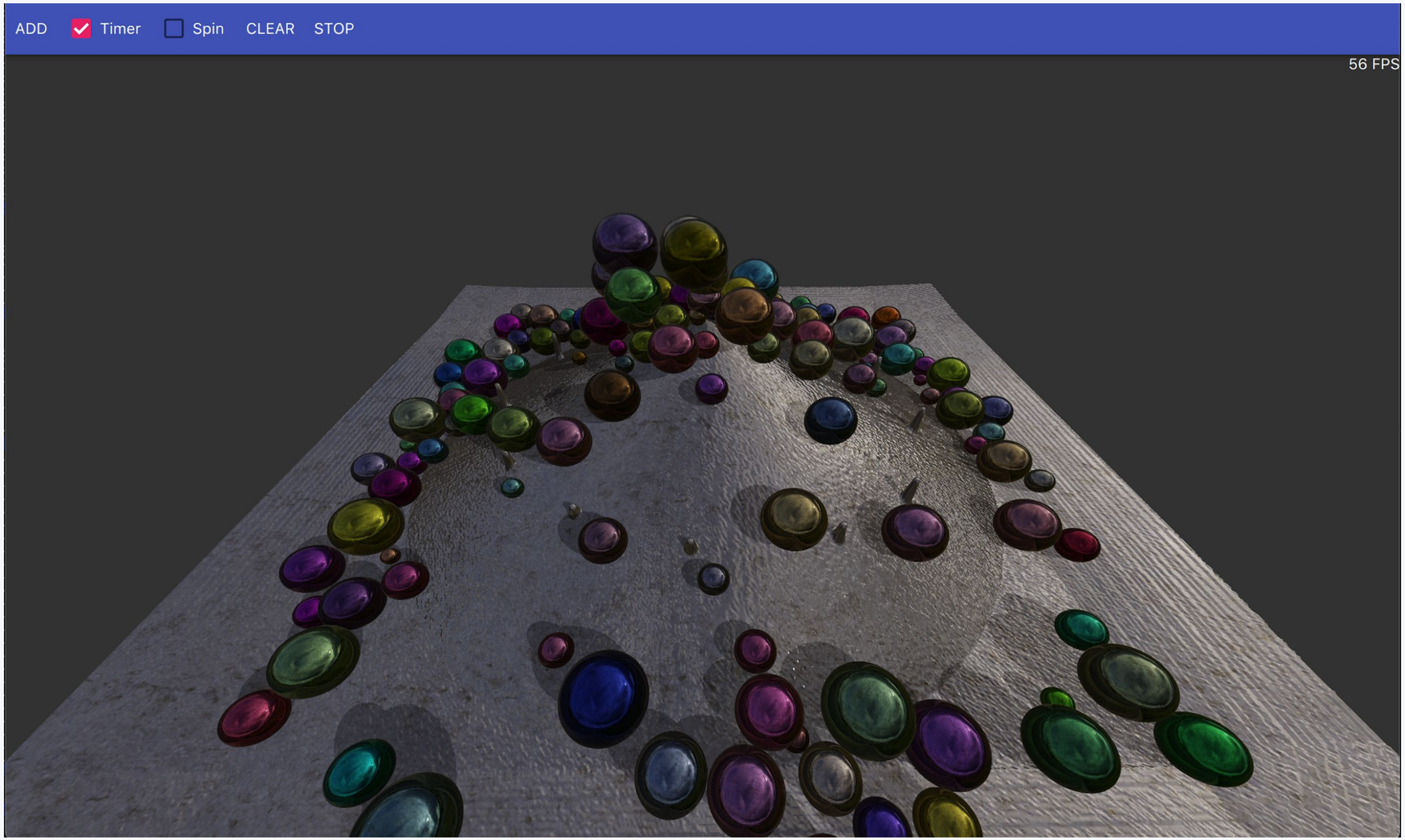
Qt 3D



QtCon BR

Source: blog.qt.io

Simple [Demo]



QtCon BR

Simple [Demo]

```
import QBullet 1.0 as Bullet
...
Bullet.DiscreteDynamicsWorld {
    ...
    gravity: Qt.vector3d(0, -9.8, 0)
}
```

```
...
BulletTools.Sandbox {
    heightmap: "qrc:/resources/heightmap.png"
    width: 250
    wallHeight: 50
    friction: 0.9
    restitution: 0.5
    ...
}
```

QtCon BR

Simple [Demo]

```
import QBullet 1.0 as Bullet
...
Bullet.SphereShape {
    id: sphereShape
    ...
}

Bullet.RigidBody {
    id: ballBody
    collisionShape: sphereShape
    ...
}

Ball {
    matrix: ballBody.matrix
    ...
}
...
```

```
// Ball
…
Entity {
    property alias matrix: transform.matrix

    Material { id: material }
    SphereMesh { id: mesh }
    Transform { id: transform }

    components: [ material, mesh, transform ]
}

```

Complex [Demo]

Obrigado!

perguntas?

QtCon BR

@patrickelectric

ArduPilot/ardupilot

bluerobotics/ping-viewer

bluerobotics/ping-protocol

mavlink/qgroundcontrol

patrickelectric/Conway-game-of-life-in-pyside2

patrickelectric/Qt3D-Konqi-Simulator

patrickelectric/SLAM_Qt

robotadasufsc/Exposer

robotadasufsc/Exposer-Gui

Williangalvani/provant-groundstation

csaga77/bullet-physics-qml-plugin

Patrick Pereira
patrick@bluerobotics.com
patrickjp@kde.org

@patrickelectric

@patrickelectric

https://patrickelectric.work